

---

Subject: [PATCH 12/12] fuse: optimize \_\_fuse\_direct\_io()  
Posted by [Maxim Patlasov](#) on Fri, 26 Oct 2012 15:50:36 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

\_\_fuse\_direct\_io() allocates fuse-requests by calling fuse\_get\_req(fc, n). The patch calculates 'n' based on iov[] array. This is useful because allocating FUSE\_MAX\_PAGES\_PER\_REQ page pointers and descriptors for each fuse request would be waste of memory in case of iov-s of smaller size.

Signed-off-by: Maxim Patlasov <[mpatlasov@parallels.com](mailto:mpatlasov@parallels.com)>

---

fs/fuse/file.c | 25 ++++++-----  
1 files changed, 21 insertions(+), 4 deletions(-)

diff --git a/fs/fuse/file.c b/fs/fuse/file.c  
index fad8c7b..1a9ae5a 100644  
--- a/fs/fuse/file.c  
+++ b/fs/fuse/file.c  
@@ -1088,14 +1088,14 @@ static int fuse\_get\_user\_pages(struct fuse\_req \*req, struct iov\_iter  
\*ii,  
 return 0;  
 }  
  
- while (nbytes < \*nbytesp && req->num\_pages < FUSE\_MAX\_PAGES\_PER\_REQ) {  
+ while (nbytes < \*nbytesp && req->num\_pages < req->max\_pages) {  
 unsigned npages;  
 unsigned long user\_addr = fuse\_get\_user\_addr(ii);  
 unsigned offset = user\_addr & ~PAGE\_MASK;  
 size\_t frag\_size = fuse\_get\_frag\_size(ii, \*nbytesp - nbytes);  
 int ret;  
  
- unsigned n = FUSE\_MAX\_PAGES\_PER\_REQ - req->num\_pages;  
+ unsigned n = req->max\_pages - req->num\_pages;  
 frag\_size = min\_t(size\_t, frag\_size, n << PAGE\_SHIFT);  
  
 npages = (frag\_size + offset + PAGE\_SIZE - 1) >> PAGE\_SHIFT;  
@@ -1131,6 +1131,23 @@ static int fuse\_get\_user\_pages(struct fuse\_req \*req, struct iov\_iter \*ii,  
 return 0;  
 }  
  
+static inline int fuse\_iter\_npages(const struct iov\_iter \*ii\_p)  
+{  
+ struct iov\_iter ii = \*ii\_p;  
+ int npages = 0;  
+  
+ while (iov\_iter\_count(&ii) && npages < FUSE\_MAX\_PAGES\_PER\_REQ) {  
+ unsigned long user\_addr = fuse\_get\_user\_addr(&ii);  
+ unsigned offset = user\_addr & ~PAGE\_MASK;

```

+ size_t frag_size = iov_iter_single_seg_count(&ii);
+
+ npages += (frag_size + offset + PAGE_SIZE - 1) >> PAGE_SHIFT;
+ iov_iter_advance(&ii, frag_size);
+ }
+
+ return min(npages, FUSE_MAX_PAGES_PER_REQ);
+}
+
static ssize_t __fuse_direct_io(struct file *file, const struct iovec *iov,
    unsigned long nr_segs, size_t count,
    loff_t *ppos, int write)
@@ -1145,7 +1162,7 @@ static ssize_t __fuse_direct_io(struct file *file, const struct iovec *iov,

    iov_iter_init(&ii, iov, nr_segs, count, 0);

- req = fuse_get_req(fc, FUSE_MAX_PAGES_PER_REQ);
+ req = fuse_get_req(fc, fuse_iter_npages(&ii));
    if (IS_ERR(req))
        return PTR_ERR(req);

@@ -1180,7 +1197,7 @@ static ssize_t __fuse_direct_io(struct file *file, const struct iovec *iov,
    break;
    if (count) {
        fuse_put_request(fc, req);
- req = fuse_get_req(fc, FUSE_MAX_PAGES_PER_REQ);
+ req = fuse_get_req(fc, fuse_iter_npages(&ii));
        if (IS_ERR(req))
            break;
    }

```

---